Hybrid Models

Role based Access Control
Objective

- Define/Understand
  - Chinese Wall Model
  - Role-based Access Control model
- Overview the secure interoperation issue
Hybrid Policies
Chinese Wall Model

- **Supports confidentiality and integrity**
  - Information flow between items in a Conflict of Interest set
  - Applicable to environment of stock exchange or investment house

- **Models conflict of interest**
  - *Objects*: items of information related to a company
  - *Company dataset* (CD): contains objects related to a single company
    - Written \( CD(O) \)
  - *Conflict of interest class* (COI): contains datasets of companies in competition
    - Written \( COI(O) \)
    - Assume: each object belongs to exactly one \( COI \) class
Example

Bank COI Class
- Bank of America
- PNC Bank
- Citizens Bank

Gasoline Company COI Class
- Shell Oil
- Standard Oil
- ARCO
- Union‘76
CW-Simple Security Property
(Read rule)

- CW-Simple Security Property
  - $s$ can read $o$ iff any of the following holds
    - $\exists o' \in PR(s)$ such that $CD(o') = CD(o)$
    - $\forall o', o' \in PR(s) \Rightarrow COI(o') \neq COI(o)$, or
    - $o$ has been “sanitized”
      ($o' \in PR(s)$ indicates $o'$ has been previously read by $s$)

- Public information may belong to a CD
  - no conflicts of interest arise
  - Sensitive data sanitized
Writing

- Alice, Bob work in same trading house
- Alice can read BankOfAmercia’s CD,
- Bob can read CitizensBanks’s CD,
- Both can read ARCO’s CD
- Alice could write to ARCO’s CD,
  - what is a problem?
CW-*-Property (Write rule)

- CW-*- Property
  - \textit{s} can \textit{write} \textit{o} iff the following holds
    - The CW-simple security condition permits \textit{S} to read \textit{O}.
    - For all unsanitized objects \textit{o}', \textit{s} can read \textit{o}' \Rightarrow CD(o') = CD(o)

- Alice can read both CDs
  - Is Condition 1 met?
  - She can read unsanitized objects of BankOfAmercia, hence condition 2 is false
    - Can Alice write to objects in ARCO’s CD?
Role-Based Access Control
Role Based Access Control (RBAC)

- Access control in organizations is based on “roles that individual users take on as part of the organization”
  - Access depends on function, not identity
    - Example:

      Allison is bookkeeper for Math Dept. She has access to financial records. If she leaves and Betty is hired as the new bookkeeper, Betty now has access to those records. The role of “bookkeeper” dictates access, not the identity of the individual.

- A role is “is a collection of permissions”
RBAC

Users  | Permission
\hline
u_1     | o_1
u_2     | o_2
u_n     | o_m

Total number Of assignments Possible?

Total number Of assignments Possible?
RBAC (NIST Standard)

What model entity would relate to the traditional notion of subject?

Total number of subjects possible?

Role vs Group?
Core RBAC (relations)

- Permissions = \(2^{\text{Operations} \times \text{Objects}}\)

- \(\text{UA} \subseteq \text{Users} \times \text{Roles}\)

- \(\text{PA} \subseteq \text{Permissions} \times \text{Roles}\)

- \(\text{assigned\_users}: \text{Roles} \rightarrow 2^{\text{Users}}\)

- \(\text{assigned\_permissions}: \text{Roles} \rightarrow 2^{\text{Permissions}}\)

- \(\text{Op}(p)\): set of operations associated with permission \(p\)

- \(\text{Ob}(p)\): set of objects associated with permission \(p\)

- \(\text{user\_sessions}: \text{Users} \rightarrow 2^{\text{Sessions}}\)

- \(\text{session\_user}: \text{Sessions} \rightarrow \text{Users}\)

- \(\text{session\_roles}: \text{Sessions} \rightarrow 2^{\text{Roles}}\)

\[
\text{session\_roles}(s) = \{ r | (\text{session\_user}(s), r) \in \text{UA} \}\]

- \(\text{avail\_session\_perms}: \text{Sessions} \rightarrow 2^{\text{Permissions}}\)
RBAC with Role Hierarchy

- Users
- Roles
- Sessions
- Operations
- Objects

Relationships:
- User to Role: UA
- Role to Permission: PA
- User to Session: user_sessions (one-to-many)
- Role to Session: role_sessions (many-to-many)
- Role Hierarchy (RH)
RBAC with General Role Hierarchy

- **authorized_users**: Roles $\rightarrow 2^{\text{Users}}$
  
  $\text{authorized_users}(r) = \{u \mid r' \geq r \& (r', u) \in UA\}$

- **authorized_permissions**: Roles $\rightarrow 2^{\text{Permissions}}$
  
  $\text{authorized_permissions}(r) = \{p \mid r \geq r' \& (p, r') \in PA\}$

- RH $\subseteq$ Roles x Roles is a partial order
  
  - called the inheritance relation
  - written as $\geq$

  $(r_1 \geq r_2) \rightarrow \text{authorized_users}(r_1) \subseteq \text{authorized_users}(r_2)$ & $\text{authorized_permissions}(r_2) \subseteq \text{authorized_permissions}(r_1)$

What do these mean?
Example

authorized_users(Employee)?
authorized_users(Administrator)?
authorized_permissions(Employee)?
authorized_permissions(Administrator)?
Constrained RBAC

Static Separation of Duty

Role Hierarchy (RH)

Users $\rightarrow$ Roles $\rightarrow$ Operations $\rightarrow$ Objects

Permissions

Dynamic Separation of Duty

user_sessions (one-to-many)

Sessions $\rightarrow$ Roles $\rightarrow$ Users $\rightarrow$ UA $\rightarrow$ PA
Static Separation of Duty

- $SSD \subseteq 2^{\text{Roles}} \times N$
- In absence of hierarchy
  - Collection of pairs $(RS, n)$ where $RS$ is a role set, $n \geq 2$
    - $\forall (RS, n) \in SSD, \forall t \subseteq RS$
      - $|t| \geq n \rightarrow \bigcap_{r \in t} \text{assigned\_users}(r) = \emptyset$
- In presence of hierarchy
  - Collection of pairs $(RS, n)$ where $RS$ is a role set, $n \geq 2$
    - $\forall (RS, n) \in SSD, \forall t \subseteq RS$
      - $|t| \geq n \rightarrow \bigcap_{r \in t} \text{authorized\_users}(r) = \emptyset$

Describe!
Dynamic Separation of Duty

- \( DSD \subseteq 2^{\text{Roles}} \times N \)
  - Collection of pairs \((RS, n)\) where \(RS\) is a role set, \(n \geq 2\);
    - A user cannot activate \(n\) or more roles from \(RS\)
  - What is the difference between SSD or DSD containing:
    \((RS, n)\)?
      - Consider \((RS, n) = (\{r_1, r_2, r_3\}, 2)\)?
      - If SSD – can \(r_1, r_2\) and \(r_3\) be assigned to \(u\)?
      - If DSD – can \(r_1, r_2\) and \(r_3\) be assigned to \(u\)?
Can we represent BLP using RBAC?
Advantages of RBAC

- Allows Efficient Security Management
  - Administrative roles, Role hierarchy
- Principle of least privilege allows minimizing damage
- Separation of Duty constraints to prevent fraud
- Allows grouping of objects / users
- Policy-neutral - Provides generality
- Encompasses DAC and MAC policies
RBAC’s Benefits

<table>
<thead>
<tr>
<th>TASK</th>
<th>RBAC</th>
<th>NON-RBAC</th>
<th>DIFFERENCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assign existing privileges to new users</td>
<td>6.14</td>
<td>11.39</td>
<td>5.25</td>
</tr>
<tr>
<td>Change existing users’ privileges</td>
<td>9.29</td>
<td>10.24</td>
<td>0.95</td>
</tr>
<tr>
<td>Establish new privileges for existing users</td>
<td>8.86</td>
<td>9.26</td>
<td>0.40</td>
</tr>
<tr>
<td>Termination of privileges</td>
<td>0.81</td>
<td>1.32</td>
<td>0.51</td>
</tr>
</tbody>
</table>
Cost Benefits

- Saves about 7.01 minutes per employee, per year in administrative functions
  - Average IT admin salary - $59.27 per hour
  - The annual cost saving is:
    - $6,924/1000;
    - $692,471/100,000

How do we get this?
Policy Composition
Problem: *Consistent Policies*

- Policies defined by different organizations
  - Different needs
  - But sometimes subjects/objects overlap
- Can all policies be met?
  - Different categories
    - Build lattice combining them
  - Different security levels
    - Need to be *levels* – thus must be able to order
  - What if different DAC and MAC policies need to be integrated?
Secure Interoperability

- **Principles of secure interoperation** [Gong, 96]
  
  **Principle of autonomy**
  - If an access is permitted within an individual system, it must also be permitted under secure interoperation.

  **Principle of security**
  - If an access is not permitted within an individual system, it must not be permitted under secure interoperation.

- Interoperation of secure systems can create new security breaches.
Secure Interoperability (Example)

$F_{12} = \{a, b\}$  

$F_{12} - \text{permitted access between systems 1 and 2}$

$F_{12} = \{a, b, c, d\}$

(1) $F_{12} = \{a, b, d\}$  
Direct access  

(2) $F_{12} = \{c\}$  
Indirect access
Summary

- Integrity policies
  - Level based and non-level based
- Chinese wall is a dynamic policy
  - Conflict classes
- RBAC – several advantages
  - based on duty/responsibility/function
  - Economic benefits as well as diversified